grams. Introducing structured high-level languages such as Micropascal reduces the need to manually calculate the explicit addresses of BRANCH and CALL instructions and further improves software quality. But the most important characteristic of an HLL such as Micropascal, Pascal, or Modula-2 is the existence of a syntax graph (see Figure 1 again). Such a graph is deterministic, according to Wirth's definition, and therefore one can systematically derive a parser, i.e., a program for the automatic detection of syntax errors.

Our second motive was didactic. It seemed important to us to be able to offer a university course integrating basic hardware concepts (hardwired switching circuits) with basic software concepts (programming or microprogramming). Our course was based on acceptance of the fundamental equivalences between

- combinational switching circuits (hardware) and binary decision trees (software), and
- finite automata or sequential systems (hardware) and program schemas (software).\(^7\)\(^8\)

**Toward a hardware autocompiler.** A last word about the architecture of our microcompiler: if the DO... instruction that realizes the microcontrol state MC\(_{256}\) is coded (i.e., is a vertical microinstruction), we obtain a binary decision machine identical to the one that executes the LLL. In other words, we obtain a BD machine that works only with 16-bit words. As a result, the LLL program of the control part of our compiler can be transformed or rewritten with an HLL introduced and transformed by the microcompiler itself. We therefore obtain a hardware autocompiler that allows us, after some LLL bootstrapping, to complete or improve our original compiler microprogram by using an HLL.

**Execution speed.** One HLL line produces, at most, one LLL instruction, which finally needs one clock pulse to be executed. Comparing this with the actual execution of a standard IF...THEN...ELSE... Pascal instruction in a classical microprocessor shows that we have simplified the interpretation process radically and improved the execution speed dramatically. 

**Acknowledgments**

I wish to thank André Stauffer of the Swiss Federal Institute of Technology, Lausanne, for designing the compilation algorithm and realizing a first version of the compiler, which was completely hardwired. I also thank Mark P. McPherson and Steffen Bartschat of Carnegie-Mellon University for verifying and improving the French-English translation.

**References**


**Daniel A. Mange**'s biography appears in the February issue, page 41.

Questions about this article can be directed to Mange at EPFL, 16 Chemin de Bellerive, CH-1007 Lausanne, Switzerland.

---

**Reader Interest Survey**

Indicate your interest in this article by circling the appropriate number on the Reader Interest Card.

High 156  Medium 157  Low 158

---

**Correction**

In Part I of this article (February), the text appearing in the box on page 37, in the section “First example: the 2909 (or 2911),” should have read as follows:

With a four-bit \(O_{PC3,0}\), for example, one obtains the following relations:

\[ S_1 = (O_{PC2} \cdot O_{PC1} \cdot O_{PC0} + O_{PC2} \cdot O_{PC1} \cdot O_{PC0} + O_{PC2} \cdot O_{PC1} \cdot O_{PC0} \cdot x_{j}) \cdot O_{PC3} \]

\[ S_0 = (O_{PC2} \cdot O_{PC1} \cdot O_{PC0} + O_{PC2} \cdot O_{PC1} \cdot O_{PC0} \cdot x_{j}) \cdot O_{PC3} \]

\[ F_E = O_{PC0} \]

\[ P_U = O_{PC1} \]

These formulas are the same as those in Figure 2 of the box on page 37.

April 1986